

PITZ Video System

Stefan Weisse¹, Ilia Chachkhunashvili²,
 DESY, Zeuthen, Germany

¹on leave from University of Applied Sciences, Wildau, Germany

²on leave from HEPI, 380086, Tbilisi, Georgia

PITZ (**P**hoto **I**njector **T**est **F**acility **Z**euthen) is a test facility at DESY Zeuthen for research and development on laser driven electron sources for Free Electron Lasers (FEL) and linear colliders [1]. The optimisation of an electron gun is only possible based on an extended diagnostic system including a Video-system. The goal is to measure the electron beam position and the profile of the beam at different places and by different diagnostic tools along the beam line [2, 3].

1. Frame Grabbing Solution

The idea of the frame grabbing is to deliver live video to client applications in order to watch or analyse the beam. For example, we want the size of the spot and the centre position of it. For the calculations that are done, it is extremely important that all frames arrive on the client computers properly. In addition, we cannot tolerate loss of information in the video signal, so we use a loss-less compression when transferring the data over the network. We cannot transmit the data uncompressed, because the required network bandwidth would be too much. The cameras deliver the video signal in non-interlaced mode. That means, unlike television, all pixels are taken at the same time. We are running at 5 frames per second at the moment.

The frame grabber card is equipped with a 4 to 1 multiplexing unit. Only the selected camera may be read out. Actually we are running 6 cameras on two servers in triggered mode synchronous to the accelerator (gun).

After the trigger, the actually selected camera sends the frame data to the frame grabber card, which acquires the image. It is read out, compressed and send out to each connected client. The cameras operate in non-interlaced mode. The resolution of the frame is 768x574 pixels. We are grabbing just the luminosity, all grabbed frames are greyscale images with a depth of 8 bit. For acquiring the frames a special industrial frame grabber card supporting the features of the cameras is used.

The client then receives and decompresses the data and does with it whatever it wants. Unlike other TV systems [6], we are using a loss-less compression with a compression ratio of about 2:1.

If you have just a spot on the frame, the compression is about 3:1 to 4:1. This reduces the required network bandwidth. The uncompressed data rate on 5 Hz is approximately 2.1 MByte/s (5 Hz * 768 * 574 * 1 Byte) .

The data rate applies to each connected client. Using 100 MBit network and an average compression ratio of about 2:1, we can serve up to 9 simultaneous clients. Using not more than 9 clients, we can guarantee that all the data will be received successfully on all clients without losing information. Because we are using loss-less compression, the frames on client side are bit-by-bit identical to the acquired images on the server. This is important for the calculations and analysis.

In addition to the socket connection that transfers the raw data there are also some control connections. These control connections are using the TINE (Three-fold Integrated Network Environment) protocol [4, 7] to switch the camera port and transfer some additional data to simplify the server handling.

At first we implemented parts based on Windows systems. This parts consist of a grabber server and a video client, which analyses the data and displays them in various modes. The Windows part was successfully used in experiment runs.

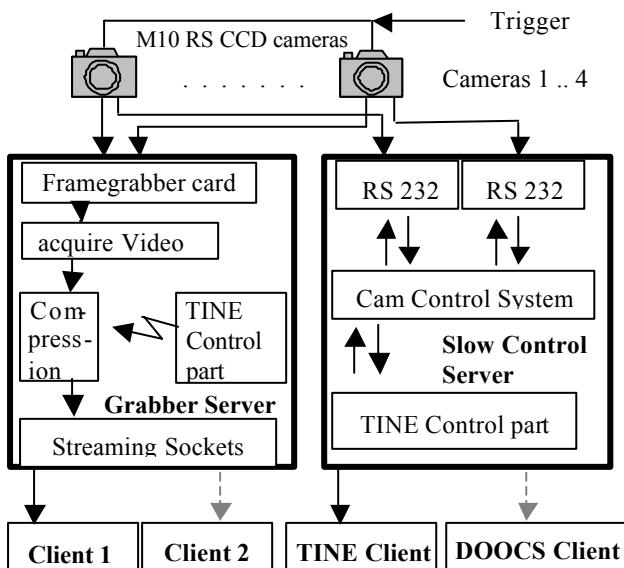


Fig. 1: Schematic Diagram of the PITZ Video System

The client was written in tight cooperation with the physicists at DESY Zeuthen. It features things like projections of the video signal, false colour mode, background subtraction, averaging frames, spot size and position calculation, selectable area of interest, xray filtering, live video mode and loading and saving of frames as well as background images.

In addition to this we have ported some functions for use with the ROOT toolkit [8], so that we are able to process the frames there. The ROOT port was incorporated into a momentum measurement and analysis system.

The next step we did was the creation of a live video viewer to be used in the control system running on UNIX. We chose the Qt class library for implementing this client. Live video is now displayed on the control monitors so that the operators can have a look what is going on without the need to start the Windows client, which is used for analysis normally.

A recipe and base classes (ANSI C++) for Unix are in preparation, so that Unix developers have the ability to get the frames, decompress them and process the video data in different types of applications.

2. RS232 Camera Control

During grabbing some properties of video cams have to be changed. As servers which control cameras are remote from the control centre a Remote Slow Control system is necessary.

The main purpose is to allow the adjustment of various parameters for a M10 RS CCD camera via a PC and the network. This means that the adjustments can be made without accessing the camera locally. It makes it easy to control the camera setup even if the camera is placed in area with restricted access. Furthermore it is possible to manage more than one camera contemporary over the network by using the client application. It is possible to save different camera setups to disk and restore it later on the same camera or duplicate the configuration for other cameras.

Under normal working conditions the configuration of the video camera is stored inside the server. But in special cases it is necessary to refresh the configuration data right from video camera because it is possible to change the configuration of cameras by using software provided by JAI [5]. On the other hand this function can be used to check the connection status of the camera (still connected, powered or off) because it is impossible to get

corresponding information via the RS232 lines, unless the server is restarted.

Two major versions of the client application have been implemented using Borland C++ Builder 6. In the first version all properties of the server have been refreshed periodically. This refresh rate was adjustable, but a "Send Button" was needed to change the settings. This design was uncomfortable, but without refresh we are not sure that properties have been changed by another operator working in parallel. But normally only one terminal was used to adjust cameras settings and we considered to implement a new version of the client, where changes will be sent to the server immediately and the settings are not refreshed by the server at the same time. Now we are working on new ergonomics for the client GUI.

The core control system of PITZ is DOOCS based [9]. The common API (Application Program Interface) of DOOCS supports the native TINE protocol and guarantees the interoperability with the grabber server and the camera control server. There are only some restrictions concerning the data types and the handling of special properties. Finally GUI's to control the grabber and the cameras are integrated into the Doocs Data Display.

Acknowledgements

The authors wish to express their gratitude to our colleagues at PITZ and DESY Zeuthen, helping with ideas and own experience. The authors thank Velizar Miltchev for providing some analysis theory and classes for the programs and Gunter Trowitzsch for constant help and support. Bert Schoeneich provided us the DOOCS GUI's and Dirk Lipka the momentum measurement application. Philip Duval, author of TINE, helped us a lot to include TINE support and to get it running properly.

References:

- [1] PITZ homepage:
<http://desyntwww.desy.de/pitz/index.html>
- [2] Juergen Baehr et al., Development of a TV Diagnostic System for the Photo Injector Test Facility at DESY Zeuthen, FEL 2001,
<http://desyntwww.desy.de/pitz/tuovi/tv2.ps>
- [3] Juergen Baehr et al., Diagnostics for the Photo Injector Test Facility at DESY Zeuthen, DIPAC 2001,
<http://desyntwww.desy.de/pitz/tuovi/grenpap.ps>

- [4] TINE protocol user documentation
(Three-fold Integrated Networking Environment)
<http://desyntwww.desy.de/tine/>

- [5] JAI Camera Solutions
<http://www.jai.com/camera/index.asp/sprog=uk>

- [6] Kay Wittenburg (DESY Hamburg, Germany), Fedor Solodovnik (IHEP Protvino, Russia), "A High Performance Multi-Protocol Network-Based Application For Video Beam Diagnostics", PCaPAC 2001,
<http://desyntwww.desy.de/PCAPAC/Proceedings/ID/141/pc97a4.doc>

- [7] P. Duval, "TINE: An Integrated Control System for HERA", PCaPAC 1999,
<http://conference.kek.jp/PCaPAC99/cdrom/paper/poster/p28.pdf>

- [8] ROOT-Toolkit by CERN
<http://root.cern.ch/>

- [9] G.Grygiel, O.Hensler, K.Rehlich
DOOCS: a Distributed Object Oriented Control System on PC's and Workstations
<http://tesla.desy.de/doocs>